Using Database Techniques for Realizing Context Aware Mobile Information Systems

Hagen Höpfner
hoepfner@acm.org
International University in Germany
School of Information Technology
Campus 3, 76646 Bruchsal, Germany

Abstract

Nearly all common information systems are based on database systems or at least use technology that is also used in database systems. In order to realize modern information systems that are on the one hand context aware and on the other hand support mobile clients, one has to consider reusing, extending and improving existing solutions. In this paper we summarize and classify the work we did so far in terms of using database techniques for realizing a context aware mobile information system, and discuss open issues.

Keywords: Mobile Information Systens, Context Aware Systems, Databases

1 Introduction and Motivation

Mobile information systems are typically classical information systems with mobile clients. However, the usage of mobile, lightweight devices as clients leads to new requirements:

- Mobile clients (MC) or their users, respectively, are mobile and the usability of the devices is restricted. So, users want the "system" to work in a context aware manner.
- Mobile clients use wireless connections in order to access the server of the information system. Thus, communication is often slow and expensive.

Classical information systems are mostly based on data base systems or at least use technology that is also used in database systems. Our idea is to reuse this technology in order to assemble context aware mobile information systems. The vision behind our work, and the idea of a scalable mobility server (SMoS) was presented in (Höpfner and Sattler, 2003b). Figure 1 shows the basic architecture of our system. Mobile clients post context aware queries (CAQ) to the server. On the server, the queries are answered using preprocessed (integrated and fragmented) data that were derived from heterogeneous data sources (e.g. databases or websites). The clients cache received data locally in a semantic manner and the server stores the answered queries in order to support the maintenance of clients' caches.
Obviously, such a system is tangential to various database techniques. In the following paper we describe and classify our approaches towards a pure database based context aware mobile information system.

2 Database Fragmentation

As described above, data from various data sources are integrated and fragmented in SMoS. That means that data, which are not context dependent, are annotated by contextual information. Many approaches

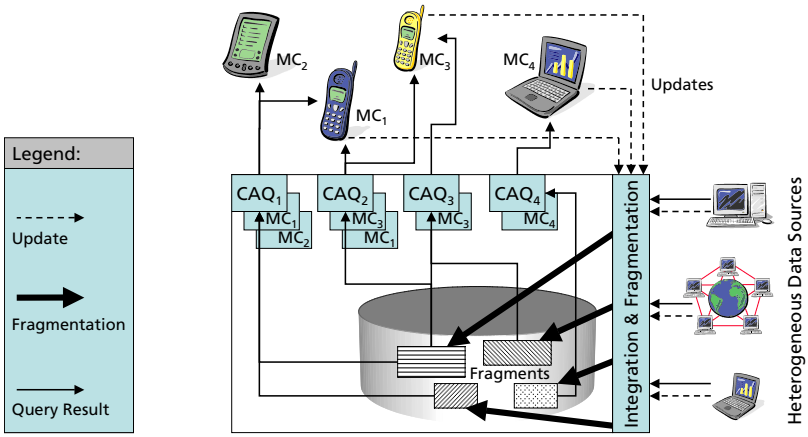

Figure 1: Architecture of the Scalable Mobility Server

reduce contextual information to location information (e.g. (Seydim et al., 2001b; Seydim et al., 2001a)) or to other subsets of possible context elements (e.g. (Prante et al., 2004)). Our approach is not restricted in this way but allows to freely model elements of the user and the device context (Höpfner and Sattler, 2003a). Nevertheless, we use locations as examples in the following. Existing context based data should be integrated into SMoS' database. Because the integration is an open issue we focus here on the fragmentation.

Fragments are views on the databases that where enhanced by additional (contextual) information (e.g. position, time), that are called server extensions $v^{se} \in \mathcal{V}_{se}$. Server extensions are vectors (e.g. two-dimensional geographic coordinates $v^{se}_{gc}(x, y)$; $x, y \in \mathbb{R}$). The parameters (in our example x and y) may be parameter-functions that compute the values of the parameters by querying the database. So, the redundant storage of data, that are already in the database, is avoided.

At the moment, the annotation is done manually but automatic fragmentation could use additional meta information about the data sources in order to calculate the server extensions. If, for example, a data source is a web site, the location of the server could be used as location annotation. Other contextual information could be extracted from the content by wrappers, but this is future work.

3 Context Aware Queries

The context aware queries in our system are based on the conjunctive queries with inequalities and self-joins, that were first presented in (Höpfner, 2005a) as predicate sequence queries (PSQ). As discussed in (Höpfner, 2005b) context aware queries (CAQ) are PSQ-queries with additional context predicates, so called replication criteria (Höpfner and Sattler, 2003a). A replication criterion is a triple and consists of a client-extension v^{ce}_{name} and of the minimal postulated and the maximal allowed tolerance of this criterion (e.g. $rk_{gc} = ((100, 200), 0, 20)$).

Answering such queries requires additional server capabilities. At first the correct fragments are selected. The decision, whether a fragment is of interest for the client or not, is done on the server using a ξ-function. If the result of this function is greater than or equal to the minimal postulated threshold and lower than or equal to the maximal allowed tolerance the respective fragment is taken into account. In our example such a ξ-function is the Euclidean distance $\xi_{gc}(v_{gc_1}, v_{gc_2}) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. However, a ξ-function must not necessarily have the properties of a distance function, e.g., does not have to be symmetric in the arguments.

The second step is to perform the PSQ-part of the CAQ on the selected fragments.

4 Client Side Caching

In order to reduce the transmission costs clients cache received data locally in a semantic manner (Höpfner and Sattler, 2003c). Therefore, queries form a query index (in form of a Trie (Fredkin, 1959; Fredkin, 1960)), that indexes the respective results. If a new query appears, we try to find correlations between the new query and the query index. If the query is included in the index, it can be answered completely locally. If a part of the required answer is in the cache, this can be reused (by performing filter queries) but, if necessary, has to be completed with data from the server (using compensatory queries). Only if no usable data is available locally, the query has to be answered in total by the server. Finding the correct correlation between cache and query and computing compensatory or filter queries are well known theoretical problems that suffer from the complexity of the algorithms used (Ren and Dunham, 1998; Ren and Dunham, 2003). However, we showed in (Höpfner, 2004) that it makes sense to delegate these issues to the server. We observed that the additional communication cost does not carry weight in non-hoarding scenarios with infrequent filter queries.

At the moment we ignore context predicates in the cache. But there are already semantic approaches that handle at least location aware queries (Ren and Dunham, 2000). However, our aim is a context aware system that is not only restricted to location awareness.

5 Stateful Database Server

One of the well known problems of information systems with mobile clients is the consistency of the "global database" (Cai et al., 1997). Replication (Höpfner, 2002) as well as caching and hoarding lead to temporary inconsistencies if data are modified offline. In our system such modification may arise from the data sources as well as from the mobile clients themselves. The integration of the updates was not realized so far, but we notify mobile clients if an update arises that affects their caches. As shown in (Höpfner, 2005a) it is necessary to check the relevance of modifications at the server site if we allow PSQ. So, the server must be stateful. In order to decide about relevancy on the mobile device one has to restrict the query language. Otherwise it might happen that operators (especially projections) "remove" data (e.g. surrogate keys) that is necessary for the relevancy check.

In (Höpfner, 2005a) we presented an approach for checking the relevance of insert, delete and update operations. The idea is similar to the query index approach. Queries form a Trie and index the IDs of the mobile clients that posted the corresponding query. Basing on the incoming update and the predicates in the Trie we calculate check queries that are performed on the database. If the query is not relevant for this predicate, we do not have to check this branch of the Trie any longer. This approach is based on the closed world assumption, that is common for database systems. Other approaches that use the open world assumption try to decide about the relevancy of modifications on the semantic level by exploring the queries. Such techniques that are presented in (Maier and Ullman, 1983), (Elkan, 1990) or (Blakeley et al., 1998) suffer from the well known restrictions of the query containment problem (Solomon, 1979) and its foundations (Rosenkrantz and Hunt III, 1980).

The kind of our check queries depends on the data model used. In (Höpfner, 2005a) we considered the common relational data model that considers relations to be sets. In (Höpfner et al., 2004) we discussed the relevance checks in multi-set semantics. First ideas for checking the relevance of updates in the case of CAQ were published in (Höpfner, 2005b), where we consider database updates as well as context updates.

6 Summary and Outlook

In this paper we gave an overview of our work on a context aware mobile information system. We described the database techniques that can be (re)used in order to realize such a system. We showed that

it is necessary to consider the wireless connections as well as mobility issues and presented first solutions for some of the problems that arise.

There are many open issues that we will address in the near future. The most urgent ones are:

- Integration of already context based data
- Integration of client updates into the server database
- Considering semantic information in the client cache

Most of these questions will raise other questions. For example, the integration of client updates will be problematic if data sources are read only but have to be updated too.

Remark: Since we reuse database techniques, our work overlaps with a lot of different research areas. Because of the space limitations, this paper does not include a real section "related work," but we referenced the most important papers in the appropriate sections. Please have a look at our referenced papers, too. We there compare our research to related works in more detail.

REFERENCES

Blakeley, J. A., Coburn, N., and Larson, P.-A. (1998). Updating derived relations: Detecting irrelevant and autonomously computable updates. In Gupta, A. and Mumick, I. S., editors, *Materialized Views*, chapter 21, pages 295–322. MIT Press, London, England.

Cai, J., Tan, K. L., and Ooi, B. C. (1997). On Incremental Cache Coherency Schemes in Mobile Computing Environments. In Gray, A. and Larson, P. Å., editors, *Proceedings of the 13th ICDE*, pages 114–123, Washington, DC, USA. IEEE Computer Society.

Elkan, C. (1990). Independence of logic database queries and update. In Rosenkrantz, D. J. and Sagiv, Y., editors, *Proceedings of the ninth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 154–160, New York, NY, USA. ACM Press.

Fredkin, E. (1959). *Trie memory*. Information Memorandum, Bolt Beranek and NewMan Inc., Cambridge, MA.

Fredkin, E. (1960). Trie memory. *Communications of the ACM*, 3(9):490–499.

Höpfner, H. (2002). Replication in Mobile Information Systems. In Schubert, S., Reusch, B., and Jesse, N., editors, *Informatik bewegt - Proceedings der 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 30.9.-3.10., Dortmund*, volume P-19 of *LNI - Proceedings*, pages 590–593, Bonn. Gesellschaft für Informatik e.V. (GI).

Höpfner, H. (2004). Serverseitige Auswertung von Indexen semantischer, clientseitiger Caches in mobilen Informationssystemen. In Dadam, P. and Reichert, M., editors, *INFORMATIK 2004 - Informatik verbindet – Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI) Band 1*, volume P-50 of *LNI*, pages 298–302, Bonn. Gesellschaft für Informatik, Köllen Druck+Verlag GmbH. in German.

Höpfner, H. (2005a). *Relevanz von Änderungen für Datenbestände mobiler Clients*. Dissertation, angenommen durch die Fakult. für Informatik der Universität Magdeburg. in German.

Höpfner, H. (2005b). Towards Update Relevance Checks in a Context Aware Mobile Information System. In Cremers, A. B., Manthey, R., Martini, P., and Steinhagte, V., editors, *INFORMATIK 2005 - Informatik LIVE! – Beiträge der 35. Jahrestagung der Gesellschaft für Informatik e.V. (GI) Band 2*, volume P-68 of *Lecture Notes in Informatics (LNI)*, pages 553–557, Bonn, Germany. Gesellschaft für Informatik, Köllen Druck+Verlag GmbH.

Höpfner, H. and Sattler, K.-U. (2003a). Semantic Replication in Mobile Federated Information Systems. In James, A., Conrad, S., and Hasselbring, W., editors, *Proceedings of the 5th EFIS*, pages 36–41, Amsterdam. Ios Press Inc.

Höpfner, H. and Sattler, K.-U. (2003b). SMoS: A Scalable Mobility Server. In James, A. and Younas, M., editors, *Poster Proceedings of 20th BNCOD*, pages 49–52. School of Mathematical and Informational Sciences; Coventry University.

Höpfner, H. and Sattler, K.-U. (2003c). Towards Trie-Based Query Caching in Mobile DBS. In König-Ries, B., Klein, M., and Obreiter, P., editors, *Post-Proceedings of the Workshop Scalability, Persistence, Transactions - Database Mechanisms for Mobile Applications*, number P-43 in LNI - Proceedings, pages 106–121, Bonn. Gesellschaft für Informatik, Köllen Druck+Verlag GmbH.

Höpfner, H., Schosser, S., and Sattler, K.-U. (2004). An Indexing Scheme for Update Notification in Large Mobile Information Systems. In Lindner, W., Mesiti, M., Türker, C., Tzikzikas, Y., and Vakali, A., editors, *Current Trends in Database Technology - EDBT 2004 Workshops: PhD, DataX, PIM, P2PDB, ClustWeb, Heraklion, Greece, March 14-18, 2004, Revised Papers*, volume 3268 of *LNCS*, pages 345–354, Berlin. Springer-Verlag.

Maier, D. and Ullman, J. D. (1983). Fragments of relations. In Stonebraker, M., editor, *Proceedings of the 1983 ACM SIGMOD international conference on Management of data*, New York, NY, USA. ACM Press.

Prante, T., Stenzel, R., Petrovic, K., and Bayon, V. (2004). Exploiting Context Histories: A Cross-Tool and Cross-Device Approach to Reduce Compartmentalization when Going Back. In Dadam, P. and Reichert, M., editors, *INFORMATIK 2004 - Informatik verbindet, Band 1, Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI)*, volume 50 of *LNI*, pages 314–318, Bonn. Gesellschaft für Informatik.

Ren, Q. and Dunham, M. H. (1998). Semantic Caching and Query Processing. Technical Report 98-CSE-04, Southern Methodist University, Departement of Computer Science and Engineering, Dallas, TX.

Ren, Q. and Dunham, M. H. (2000). Using semantic caching to manage location dependent data in mobile computing. In *Proceedings of the MobiComm2000*, pages 210–221, New York, NY, USA. ACM Press.

Ren, Q. and Dunham, M. H. (2003). Semantic caching and query processing. *Transactions on Knowledge and Data Engineering*, 15(1):192–210.

Rosenkrantz, D. J. and Hunt III, H. B. (1980). Processing Conjunctive Predicates and Queries. In Taylor, A., editor, *Proceedings of the Sixth International Conference on Very Large Data Bases*, pages 64–72, Los Altos, CA, USA. IEEE Computer Society Press.

Seydim, A., Dunham, M., and Kumar, V. (2001a). An architecture for location dependent query processing. In Tjoa, A. M. and Wagner, R., editors, *DEXA '01:Proceedings of the 12th International Workshop on Database and Expert System Applications*, pages 549–550, Washington, DC, USA. IEEE Computer Society.

Seydim, A. Y., Dunham, M. H., and Kumar, V. (2001b). Location-Dependent Query Processing. In *Proceedings of the 2nd ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pages 47–53, New York, NY, USA. ACM Press.

Solomon, M. K. (1979). Some properties of relational expressions. In *Proceedings of the 17th annual Southeast Regional Conference*, pages 111–116, New York, NY, USA. ACM Press.